High Throughput Parallel Computing (HTPC)

Dan Fraser, UChicago
Greg Thain, Uwisc

The two familiar HPC Models

- High Throughput Computing
  - Run ensembles of single core jobs
- Capability Computing
  - A few jobs parallelized over the whole system
  - Use whatever parallel s/w is on the system

- HTPC – an emerging model

Ensembles of small-way parallel jobs

(10's – 1000's)

Use whatever parallel s/w you want ☺

(It ships with the job)

Current HTPC Mechanism

- Submit jobs that consume an entire processor
  - Typically 8 cores (today)
  - Package jobs with a parallel library (schedd)
    - HTPC jobs as portable as any other job
    - MPI, OpenMP, your own scripts, …
    - Parallel libraries can be optimized for on-board memory access
  - All memory is available for efficient utilization
    - Some jobs just need more memory
- Submit jobs via WMS glidein on the OSG
  - A separate list of resources for HTPC (by VO)

https://twiki.grid.iu.edu/bin/view/Documentation/HighThroughputParallelComputing

Where we are heading

- Condor 7.6.x now supports:
  - Node draining
  - Partial node usage (e.g. 4 out of 8 cores)
  - Fixes a job priority issue with HTPC jobs
- Now being tested at Wisconsin (CHTC)
  - Plan is to create a “best practice” template to help with Condor configuration
  - Will add to RPM CE doc
    - Currently in Pacman OSG CE doc

Configuring Condor for HTPC

- Two strategies:
  - Suspend/drain jobs to open HTPC slots
  - Hold empty cores until HTPC slot is open
  - http://condor-wiki.cs.wisc.edu
- See the HTPC twiki for the straightforward PBS setup
  - https://twiki.grid.iu.edu/bin/view/Documentation/HighThroughputParallelComputing

How to submit

```
universe = vanilla
requirements = (CAN_RUN_WHOLE_MACHINE =?= TRUE)
+RequiresWholeMachine=true
executable = some job
arguments = arguments
should_transfer_files = yes
when_to_transfer_output = on_exit
transfer_input_files = inputs
queue
```

MPI on Whole machine jobs

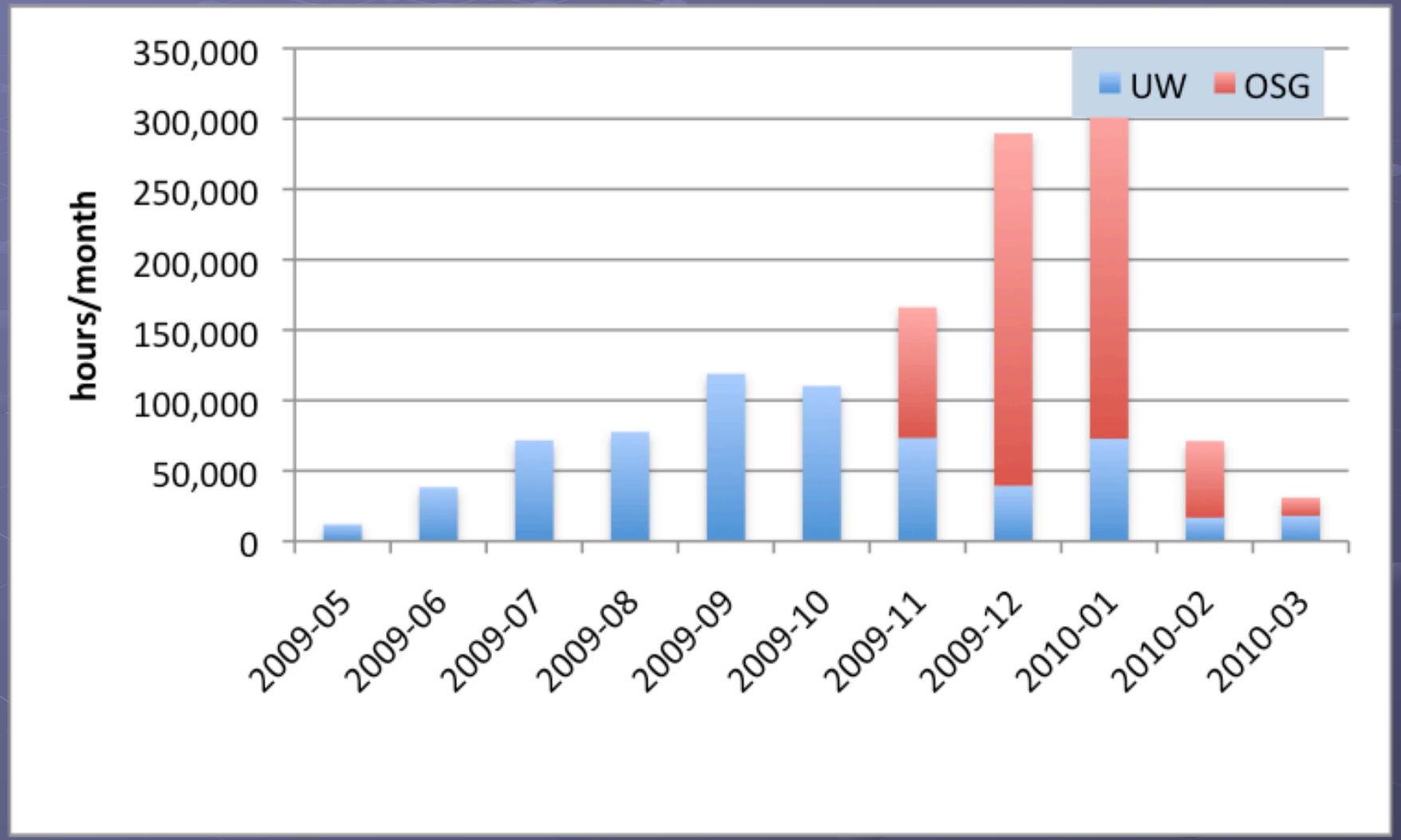
Whole machine mpi submit file

```
universe = vanilla
requirements = (CAN_RUN_WHOLE_MACHINE =?= TRUE)
+RequiresWholeMachine=true

executable = mpiexec

arguments = -np 8 real_exe

should_transfer_files = yes
when_to_transfer_output = on_exit

transfer_input_files = real_exe

queue
```

How to submit to OSG

```
universe = grid
GridResource = some_grid_host
GlobusRSL = MagicRSL
executable = wrapper.sh
arguments = arguments
should_transfer_files = yes
when_to_transfer_output = on_exit
transfer_input_files = inputs
transfer_output_files = output
queue
```

Chemistry Usage Example of HTPC



What's the magic RSL?

Site Specific

Documentation is in the OSG CE (

PBS

(host_xcount=1)(xcount=8)(queue=?)

LSF

(queue=?)(exclusive=1)

Condor

(condorsubmit=('+WholeMachine' true))

What's with the wrapper?

Chmod executable

Create output files

```
#!/bin/sh
chmod 0755 real.ex
touch output
./mpiexec -np 8 real.ex
```

- Next steps …?